|  |
| --- |
| EPAM Systems, RD Dep. |
| Практическое задание  NET.C#.11 Коллекции и обобщенные типы |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial version | Анжелика Кравчук |  |  |  |
| 1.1 | Review and corrections. | Владимир Тихон |  |  |  |
| 1.2 | Added details | Oleg |  |  |  |

**Задание 1 «Бинарное дерево поиска».**

* Разработать тип «бинарное дерево поиска» для хранения результатов тестов, выполненных студентами, и их предоставления в упорядоченом виде.
* Предусмотреть возможность хранения данных любого типа, поддерживающего сравнения.
* Информация о студенте может содержать поля, хранящие имя студента, название теста, дату его прохождения и оценку теста для данного студента.
  + Реализовать обобщенный стандартный интерфейс, чтобы информацию о студенте можно было сравнить с информацией о другом студенте.
* Предусмотреть следующие конструкторы «дерева»:
  + Который принимает объект стандартного обобщённого типа для сравнения двух объектов одного типа, и применял его для сравнения добавляемых в дерево элементов.
  + Который принимает объект типа стандартной обобщенной коллекции (не список!), и добавлял элементы коллекции в дерево (пример смотрите в «исходниках» конструкторов типа List<T>).
  + Который принимает и коллекцию элементов для добавления в дерево, и объект с помощью, которого будет выполнять сравнения.
* Итого: если указан «компаратор», то он используется для сравнения элементов, иначе – используется «сравнение, реализованное внутри элемента».
* Созданный класс «дерево» должен реализовывать обобщённый интерфейс перечислителя. При реализации нужно применить «итератор» (реализовать с помощью yield, см. важно п.3)).
* В «дереве» реализовать метод, который возвращает объект для итерации по дереву в обратном порядке.

Например, GetReversedEnumarator()(реализовать с помощью yield, см. важно п.3))

* В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение. В приложении нужно:
  + Применить созданное «дерево» для хранения чисел из целочисленного массива, при этом в дерево они сразу должны храниться в порядке убывания.
    - Добавить ещё несколько чисел в дерево.
    - Вывести все числа «дерева» в том порядке, в котором они храняться в дереве.
  + Применить созданное «дерево» для хранения результатов прохождения тестов студентами, при этом элементы в дереве **сразу** должны храниться в порядке убывания «оценок (баллов)»» по тесту.
    - Вывести информацию о студентах в том порядке, в котором они храняться в дереве.
  + Применить созданное «дерево» для хранения строк:
    - Строки добавить из списка строк;
    - Вывести строки в обратном порядке, используя метод «дерева», который возвращает «перечислитель» для прохода «дерева» в обратном порядке.

**Дополнительное задание (!):**

* Создать класс для итерации по дереве в обратном порядке (реализовать IEnumerator<T>, см. важно п.4)).
  + Изменить в «дереве» метод, который возвращает объект для итерации по дереву в обратном порядке.

**Важно (!):**

1. Само «дерево» не должно ничего выводить на «экран» (UI).
2. Если в дереве нельзя хранить элемент (значения) null, то нужно выбрасывать соответствующее исключение при попытке добавить элемент null.
3. yield (C# Reference)

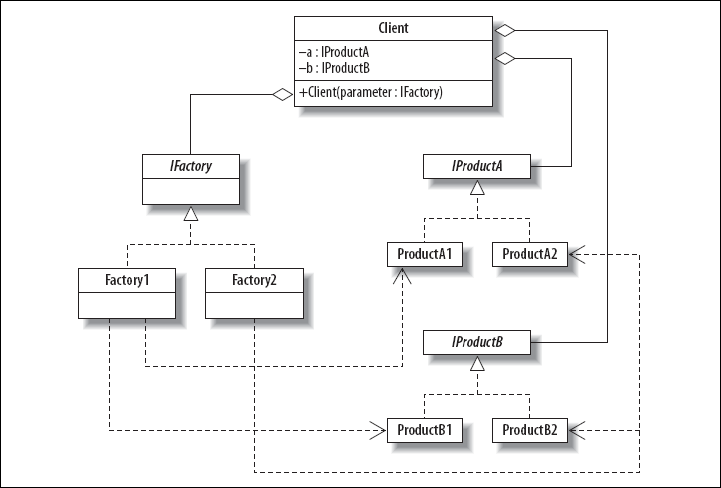
<http://msdn.microsoft.com/en-us/library/9k7k7cf0.aspx>

1. IEnumerator<T> Interface

<http://msdn.microsoft.com/en-us/library/78dfe2yb(v=vs.110).aspx>

**Задание 2 «Абстрактная фабрика».**

* Разработать иерархию классов для **произвольной предметной области** на основе следующей UML-диаграммы классов. Изучените отношения между классами и экземплярами классов.



* У интерфейса IFactory должны методы CreateProductA(), возвращающий объект типа IProductA, и CreateProductB(), возвращающий объект типа IProductB.
* **Важно (!):** Названия классов и методов должны соответствовать **выбранной предметной области**. Примеры:
  + Мебель: стол, стул, диван, шкафт,
    - Плетёная мебель, офисная мебель, мебель для столовых.
  + Организация: рабочий, начальник, директор.
    - Завод, Торговый ларёк, Оптовый склад.
  + Работа с БД: соединение с БД, выполнение запроса в БД, параметр запроса БД.
    - MS SQL Server, Oracle, SQLLite.
* **Важно (!):** Обратите внимание на конструкторы и модификаторы доступа методов, конструкторов, свойств и полей, которые указаны на UML-диаграмме.
* В коде должно быть «один тип – один файл», подробности <http://www.rsdn.ru/article/mag/200401/codestyle.xml#EIFAC>
* Для структурирования «классов» примените пространство имён, подробности <http://www.rsdn.ru/article/mag/200401/codestyle.xml#E1DAC>

Framework Design Guidelines: Names of Namespaces

<http://msdn.microsoft.com/en-us/library/vstudio/ms229026.aspx>

* + Должно быть общее пространство имён. В нём должны быть базовые интерфейсы. Пример: System.Data. **(!)**
  + И «вложенные» пространства имён для конкретных объектов. Примеры: System.Data.SqlClient, System.Data.OracleClient.
* В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.